A Framework for Linking Projects and Project Management Methods

Tony Dale, Neville Churcher, and Warwick Irwin

Software Engineering and Visualisation Group,
Computer Science and Software Engineering Department,
University of Canterbury
{Tony.Dale, Neville.Churcher, Warwick.Irwin}@canterbury.ac.nz
http://www.cosc.canterbury.ac.nz/research/RG/svg/index.html

Abstract. Software development processes such as the Waterfall process and Extreme Programming are project management methods (PMMs) which are well known and widely used. However, conventional project management (PM) lacks the process concepts expressed in PMMs, and the connection between PMMs and PM is not much explored in the literature.

We present data models for PM and PMM, in a framework that can articulate the PM–to–PMM relationship, illustrating with simple examples. A java/XML implementation of this framework can create and then revise a "PMM–aware" project, conforming to a specified PMM. In terms of the framework, we describe a simple project data visualization and associated method that can be used to synthesize a PMM for a project instance that was initially created without reference to any PMM.

Keywords: Project management, software engineering, project management methods, project processes, data modelling, XML, visualization.

1 Introduction

Conventional PM attempts to manage tasks and resources as closely as possible to a predefined, static plan.

However, a project plan doesn't indicate how its particular tasks or resources were created, except perhaps for a textual description. This is because the PM domain has no concept of why a task or resource appears in the plan. PMMs have this descriptive power because they use process concepts to formalise the specialist knowledge which any real project requires; when creating or changing. a project plan. Applying PMMs and process concepts to conventional PM creates a fundamental problem: PMMs dynamically change projects, but conventional PM is static, and has difficulty tolerating change.

It is possible to create data models for both PM and PMMs, and to link these models together in a unified framework. This framework has the power to express complicated PMM ideas but is simple and logical to apply to PM data: project tasks and resources have PMM concepts added to them as simple

F. Bomarius and S. Komi-Sirviö (Eds.): PROFES 2005, LNCS 3547, pp. 84–97, 2005.

attributes, for example. This strategy makes it possible to move backwards and forwards between the PM and PMM domains, so that a project can be created and changed according to a PMM, yet viewed with existing PM software tools.

The motivation for creating this framework is to provide a bridge between the process–dominated world of PMMs and the plan–dominated world of PM. We hope to encourage the more widespread use of process concepts and PMMs in conventional PM, and to augment PMMs with the project history and context available using conventional PM tools.

The rest of the paper is set out as follows: Section 2 introduces concepts of conventional project management (PM) as it is widely practised, and introduces a data model for PM software tools. We show in Sect. 3 how the specialist knowledge that all projects make use of can be formalized as a Project Management Method (PMM), and create a data model for describing PMMs in Sect. 4. In Sect. 5 we create a framework for linking the PM and PMM data models, and using this framework Sect. 6 illustrates how a PMM is applied to produce a project instance. Section 7 uses the framework to create a simple visualization and a heuristic method to derive a PMM from raw project data, and illustrates a scenario for applying process improvement to the derived PMM. Section 8 concludes our discussion.

2 Background

Large–scale project management has been practised for centuries, to the extent that Burbridge [1] says

> one hallmark of civilization is the ability to engage in group activities for the execution of major projects, be they tombs and temples or manned flights into space.

A crucial change has been, from the 19th century onwards, the increasing importance of time and cost in project management. Projects must be completed on time and within budget, according to comprehensive project plans, for reasons such as increasing company profitability or reducing expenditure of public funds. Unfortunately, the failure rate of projects in certain fields such as Software Engineering is very high, with the Standish Group reporting in 2001 that only 28% of IT projects completed on time and within budget [2], although this is an increase from the 16% of 1994 [3]. There is thus a motivation to provide more powerful tools for project management.

Conventional project management is encapsulated in the following five steps, as exemplified by Lewis [4]:

1. define the problem
2. plan the project
3. execute the plan
4. monitor and control progress against the plan
5. close the project.